REMARKS

This amendment is filed in response to the Final Office action mailed March 27, 2010, with Request for Continued Examination (RCE) filed herewith on even date. All rejections and objections are respectfully traversed.

Claims 3 – 13, 15, 17 – 19 and 27 – 29 are pending in this case.

Claims 1 – 2, 16, 20, and 24 – 26 have been cancelled without prejudice.

Claims 3, 12 – 13, 15, and 17 have been amended.

Claims 27 – 29 have been added.

Interview Summary

On June 28, 2010 and July 20, 2010 the Applicant's attorney conducted telephone interviews with the Examiner. The Applicant thanks the Examiner for his time. Proposed claim amendments, specifically new claim 28, and the cited references Becker-Szendy et al., U. S. Patent No. 7,243,089 (hereinafter "Becker-Szendy"), Kazar et al., U.S. Patent No. 6,868,417 (hereinafter "Kazar"), Van Hoff et al., U.S. Patent No. 5,761,421 (hereinafter "Van Hoff"), and Shu et al., U.S. Patent No. 7,555,772 (hereinafter "Shu") were discussed. The Examiner stated that the substance of the new claims, specifically that of new claim 28, were novel over the prior art of record and that a new search would be performed.

Claim Rejection – 35 USC §103

At pages 2 – 14 of the Final Office Action, claims 1 – 3, 6, 12 – 13, 15 – 17, 20, and 24 – 26 were rejected under 35 U.S.C. §103(a) over Becker-Szendy, in view of Kazar, in further view of Van Hoff, in further view of Shu.

The Applicant notes that independent claims 1, 16, 20, and 24 – 26 have been cancelled. As such, the Applicant will address the rejection with respect to new independent claims 27 – 29.

Applicant's claimed invention, as set forth by new independent claim 27, recites:

27. (New) A system comprising:
a processor;
a memory coupled to the processor;

a storage operating system resident in the memory and executed by the processor, the storage operating system implementing a file system configured to provide storage service of information stored on the system;

a plurality of network interfaces configured to process received block-based protocol data access requests, each network interface assigned to one or more network addresses, each network interface further assigned an identifier that binds the network interface to an address space that includes the one or more network addresses; and

a plurality of context data structures stored in the memory and containing configuration information to establish a plurality of instances of virtual servers executed by the processor, each virtual server allocated resources that include a partitioning of the network interfaces and assigned network addresses to establish a distinct security domain for that virtual server that enables controlled access to the allocated network interfaces and assigned network addresses, each virtual server further configured to share access to the file system to service the block-based protocol data access requests by converting the block-based protocol data access requests to appropriate file system data requests when providing the storage service of the information stored on the system.

Becker-Szendy describes a technique for federating and migrating data in a file system using virtual servers. *See* Becker-Szendy, col. 1, lines 10 – 17. Specifically, Becker-Szendy federates a local file system into a distributed file system, while preserving local access to the existing data in the local file system. *See* Becker-Szendy, col. 2, lines 52 – 54. "Unlike most file systems, meta-data and data are stored separately in the storage tank system. The server manages meta-data comprising the location of the blocks of each file/object on shared storage." (Emphasis added). *See* Becker-Szendy, col. 3, lines 2 – 5. Further, Becker-Szendy states that metadata includes "the directory tree and the attributes of objects such as files and directories.... Typical attributes comprise … security related attributes (i.e., the identity of the owner of the object and a description of what the owner or other parties may do to the object)." *See* Becker-Szendy, col. 7, 42 – 48.

Kazar describes technique for handling file level and block level remote file accesses using the same server. *See* Kazar, Abstract. Specifically, the environment includes a network file server combined with a network block protocol server, with both servers implemented on top of inode layer abstraction. *See* Kazar, col. 3, lines 21 – 23.

With respect to block level services, a block login operation passes a user ID and password and authenticates a particular user. Based upon the user, the system chooses a specific file system to which the user's block read and write operations will be applied. *See* Kazar, col. 9, line 61 – col. 10, line 1.

Van Hoff describes a technique for establishing peer-to-peer communication between computers of the same security domain. *See* Van Hoff, Abstract. During this establishment, security measures can be taken. *See* Van Hoff, col. 4, lines 35 – 36. Specifically, when establishing the peer-to-peer communication, a first virtual machine can verify that a reply packet was in fact sent by a second virtual machine by using security measures associated with a security domain that server S1, the first virtual machine, and the second virtual machine belong to. *See* Van Hoff, col. 4, lines 38 – 45. After receiving and processing the reply packet, the first virtual machine sends an acknowledgement message to the second virtual machine and the peer-to-peer communication is established. *See* Van Hoff, col. 4, lines 48 – 51.

Shu describes a technique for screening incoming packets and determining when a tunnel is torn down so that a firewall may clear an associated firewall session, thereby preventing incoming packets associated with the cleared firewall session from passing through. *See* Shu, Abstract. Further, Shu states that a firewall can be partitioned into multiple virtual systems. *See* Shu, col. 8, lines 55 – 57. Each virtual system, that is a partition of a firewall, is a unique security domain and can be managed by administrators who can individualize (e.g., including setting up address books and policies) the security protections for the given domain. *See* Shu, col. 8, lines 57 – 60.

The Applicant respectfully submits that a combination of Becker-Szendy, Kazar, Van Hoff, and Shu does not teach or suggest the Applicant's claimed "***a plurality of context data structures stored in the memory and containing configuration information to establish a plurality of instances of virtual servers executed by the processor, each virtual server allocated resources that include a partitioning of the network interfaces and assigned network addresses to establish a distinct security domain for that virtual server that enables controlled access to the allocated network interfaces and assigned network addresses, each virtual server further configured to share access to the file***

system to service the block-based protocol data access requests by converting the block-based protocol data access requests to appropriate file system data requests when providing the storage service of the information stored on the system."

The Applicant's claimed technique stores a plurality of context data structures containing configuration information to establish a plurality of instances of virtual servers. **Each vfiler is allocated resources that include a partitioning of network interfaces and assigned network addresses to establish a distinct security domain for that virtual server**. This allocation enables each virtual server to have controlled access to its allocated network interfaces and assigned network addresses. Further, the virtual servers **share access to the file system that services block-based protocol data access requests by converting the block-based protocol data access requests to appropriate file system data requests**.

As an illustrative example, consider the following. A context data structure of a first vfiler ensures that users or clients of a first security domain can use a first set network interfaces and network addresses (e.g., the allocated resources) when issuing requests to access a first subset of storage resources on a shared storage appliance. Similarly, the context data structure of a second vfiler ensures that clients of a second security domain may use a second set of network interfaces and network addresses (e.g., the allocated resources), that are distinct from the allocated resources dedicated to the first vfiler, to access a second subset of storage resources. For example, the first vfiler may be allocated network addresses 1 – 10 and have access to the file system. Similarly, the second vfiler may be allocated network addresses 22 – 40 and also have access to the file system. Advantageously, clients associated with the first vfiler and the first security domain are unaware of the "presence" of the second vfiler and the second security domain. That is, each vfiler has its own dedicated context data structure that enables controlled access to the allocated resources that include a partitioning of network interfaces and assigned network addresses to establish a distinct security domain, while allowing the virtual servers to share access to the file system that services block-based protocol data access requests by converting the block-based protocol data access requests to appropriate file system data requests.

The Applicant respectfully submits that Becker-Szendy fails to address these aspects of the Applicant's claim. Instead, Becker-Szendy describes a technique for federating and migrating data using virtual servers. *See* Becker-Szendy, col. 1, lines 10 – 16. Specifically, Becker-Szendy states that "a virtual storage tank server and a virtual object storage server on top of the local file system to make the local file system appear as both a storage tank node and an object based storage server to a storage tank system." *See* Becker-Szendy, col. 3, lines 51 – 54. "Data accesses go through the virtual object storage server, and not through the virtual storage tank server. It should be clear that the storage tank server provides metadata information to the client, who then accesses the data directly from the object based storage server." *See* Becker-Szendy, col. 3, lines 55 – 60. Becker-Szendy makes no mention of allocating resources that include **a partitioning of network interfaces and assigned network addresses to each virtual server to establish a distinct security domain for that virtual server** to enable each virtual server controlled access to its allocated network interfaces and assigned network addresses. Said differently, Becker-Szendy makes no mention of a plurality of virtual tank servers where each virtual tank server is allocated network interfaces and assigned network addresses to establish a distinct security domain. Further, Becker-Szendy makes no mention of a plurality of virtual tank servers (that were each allocated network interfaces and assigned network addresses) **sharing access to a file system to service the block-based protocol data access requests through conversion to appropriate file system data requests**. As such, the Applicant respectfully submits that Becker-Szendy may not fairly be interpreted to teach or suggest these aspects of the Applicant's claim.

Further, the Applicant respectfully submits that the deficiencies of Becker-Szendy are not remedied by a combination with Kazar. Instead, Kazar describes a technique "For handling file level and block level remote file accesses" using the same server. *See* Kazar, Abstract. Specifically, in describing a "block level service … in terms of the inode layer operation", Kazar states that "a block_login operation passes in a user ID and a password, and authenticates the user for the service. Based upon the user, the server chooses a particular file system to which the user's block read and write operations will be applied." *See* Kazar, col. 9, line 64 – col. 10, line 1. Kazar makes no mention of

allocating resources that include **a partitioning of network interfaces and assigned network addresses to each virtual server to establish a distinct security domain for that virtual server** to enable each virtual server controlled access to its allocated network interfaces and assigned network addresses. Further, Kazar makes no mention of virtual servers (that were each allocated network interfaces and assigned network addresses) **sharing access to a file system to service the block-based protocol data access requests through conversion to appropriate file system data requests**. As such, the Applicant respectfully submits that Kazar may not fairly be interpreted to teach or suggest these aspects of the Applicant's claim.

Moreover, the Applicant respectfully submits that the deficiencies of Becker-Szendy and Kazar are not remedied by a further combination with Van Hoff. Instead, Van Hoff simply states that two virtual machines (e.g., M1 and M2) may achieve peer-to-peer communication using security provisions (e.g., a security check), wherein M1 and M2 belong to the same security domain. Specifically, Van Hoff states,

> Additional security provisions, such as the use of digital signatures or the like, may be added by underlying protocol layers of the communication software used by the virtual machines, for instance so that M1 can verify that the reply packet really was sent by M2. More generally, each of the virtual machines M1 and M2, operating on corresponding client computers, will use whatever communication security measures are associated with the security domain of which they and the server S1 are members and that would normally be used for communications between those virtual machines and the server S1. However, such additional security measures are an optional part of the operating environment in which the invention may be used. (Emphasis added). *See* Van Hoff, col. 4, lines 35 – 47.
>
> Upon receipt and processing of the reply packet P2, virtual machine M1 sends an acknowledgment message back to virtual machine M2, establishing a peer-to-peer connection between applets A1 and A2 (step 214). Thereafter, the two applets exchange messages and data (step 216) in accordance with the common security restrictions shared by the two applets. (Emphasis added). *See* Van Hoff, col. 4, lines 48 – 54.

Thus, the security check utilized between virtual machines M1 and M2 are associated with the security domain that M1, M2 and server S1 belong to. That is, the security check in Van Hoff is performed in a security domain common to virtual

machines M1 and M2 and server S1. In contrast, in the Applicant's claimed technique, each virtual server is allocated resources that include a partitioning of network interfaces and assigned network addresses to **establish a distinct security domain for that virtual server**. Said differently, in Van Hoff, virtual machine M1 and M2 are part of the same security domain while in the Applicant's technique **each virtual server has a distinct security domain**. Further, the Applicant notes that Van Hoff makes no mention of its virtual machines M1 and M2 being **allocated resources that include portioning of the network interfaces and assigned network addresses**. Further, Van Hoff's virtual machines M1 and M2 do not **share access to a file system to service the block-based protocol data access requests through conversion to appropriate file system data requests**. As such, the Applicant respectfully submits that Van Hoff may not fairly be interpreted to teach or suggest these aspects of the Applicant's claim.

Finally, the Applicant respectfully submits that the deficiencies of Becker-Szendy, Kazar, and Van Hoff are not remedied by a combination with Shu. Instead, Shu simply states that a firewall may be partitioned into multiple virtual systems, where each virtual system is a unique security domain and can be managed by administrators who may individualize security protection for the domain. *See* Shu, col. 8, lines 55 – 62. The "virtual systems" (that are a partition of a firewall) as described in Shu are not akin to the Applicant's claimed virtual servers. Specifically, Shu's "virtual systems" are not allocated resources that include **a partitioning of network interfaces and assigned network addresses to each virtual server to establish a distinct security domain for that virtual server** to enable each virtual server controlled access to its allocated network interfaces and assigned network addresses. Instead, Shu's "virtual systems" as simply partitions of a firewall. Further, Van Hoff's "virtual systems", that are partitions of a firewall, do not **share access to a file system to service the block-based protocol data access requests through conversion to appropriate file system data requests**. . As such, the Applicant respectfully submits that Shu may not fairly be interpreted to teach or suggest these aspects of the Applicant's claim.

Accordingly, Applicant respectfully urges that a combination of Becker-Szendy, Kazar, Van Hoff, and Shu is legally insufficient to render the present claims unpatentable under 35 U.S.C. 103(a) because of the Absence of the Applicant's claimed "***a plurality of context data structures stored in the memory and containing configuration information to establish a plurality of instances of virtual servers executed by the processor, each virtual server allocated resources that include a partitioning of the network interfaces and assigned network addresses to establish a distinct security domain for that virtual server that enables controlled access to the allocated network interfaces and assigned network addresses, each virtual server further configured to share access to the file system to service the block-based protocol data access requests by converting the block-based protocol data access requests to appropriate file system data requests when providing the storage service of the information stored on the system***."

At pages 2 – 14 of the Final Office Action, claims 4 – 5 and 18 – 19 were rejected under 35 U.S.C. §103(a) over Becker-Szendy, in view of Kazar, in further view of Van Hoff, in further view of Shu, in further view of Mane et al., U.S. Publication No. 2005/0050107 (hereinafter "Mane").

At pages 2 – 14 of the Final Office Action, claims 7 – 11 were rejected under 35 U.S.C. §103(a) over Becker-Szendy, in view of Kazar, in further view of Van Hoff, in further view of Shu, in further view of George et al., U.S. Patent No. 7,010,663 (hereinafter "George").

The Applicant notes that claims 4 – 5, 7 – 11, and 18 – 19 are dependent claims that depend from independent claims believed to be in condition for allowance. Accordingly, claims 4 – 5, 7 – 11, and 18 – 19 are believed to be in condition for allowance due to their dependency, as well as for other separate reasons.

Conclusion

All independent claims are believed to be in condition for allowance.

All dependent claims are dependent from independent claims which are believed to be in condition for allowance. Accordingly, all dependent claims are believed to be in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Omar M. Wadhwa/

Omar M. Wadhwa
Reg. No. 64,127
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927